Recurrent Neural Networks

Learning Outcomes

By the end of this topic, you will have achieved the following learning outcomes:

- Describe how a recurrent neural network works.
- Create a recurrent neural network in order to make a prediction.
- Distinguish a recurrent neural network from a feed-forward neural network.
- Recall the benefits and limitations of recurrent neural networks
- List the applications of recurrent neural networks.

"AI is only as good as the data that we can feed it." Paul Hofmann

Reading

Overview

There are various types of neural networks. These types include ***feed-forward neural networks***, ***recurrent neural networks***, ***convolutional neural networks*** and ***deep belief neural networks***.

In ***feed-forward neural networks***, a set of inputs are taken in then processed in the hidden layers and an output is given. In such a network, data is forgotten and the next input within the hidden layers is processed independently.

On the other hand, ***recurrent neural networks (RNNs)*** allow previous outputs to be used as inputs while having hidden states. One could think of RNNs as having memory which captures information about what has been calculated so far and uses it to calculate whatever information comes after.

Such a characteristic makes these types of neural networks suitable for modeling sequential data i.e. text written in different languages where we would want to predict

words which come after other words, or in recording of speech or video where still related pieces of data occur one after the other.

Why Recurrent Neural Networks?

The primary reason why recurrent networks are important is because they substitute for the weakness caused by feed-forward neural networks which struggle with handling sequential data. In addition, feed-forward neural networks can only consider only current input and cannot memorize previous inputs as the hidden layers are all processed independently.

With Recurrent neural networks, sequential data can be processed suitably by accepting the current input data and considering previously used inputs. This all made possible by internal memory.

How does a Recurrent Neural Network work?

A typical neural network follows the following structure:

Input → Hidden → Output

whereas a recurrent neural network follows the following structure:

Input + Previous Hidden → Hidden → Output

The two diagrams below illustrate the difference in information flow between a RNN and a feed-forward neural network.

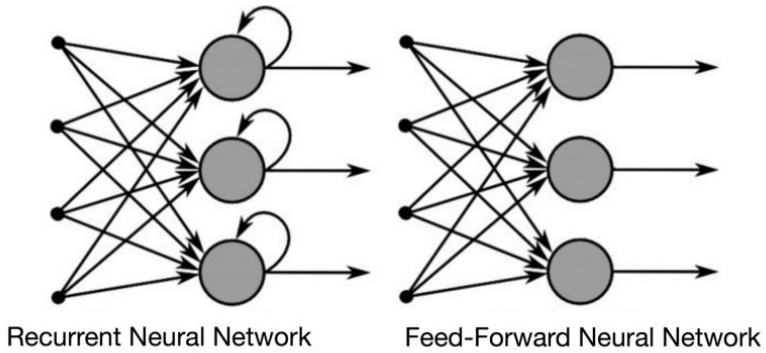


Image Source [Link]

To illustrate the how recurrent networks vs feed-forward neural work, we can use the following example:

> *Imagine we have a normal feed-forward neural network and give it the word "nairobi" as an input and it processes the word character by character. By the*

time it reaches the character "r," it has already forgotten about "i," "a" and "n", which makes it almost impossible for this type of neural network to predict which character would come next.

*A **recurrent neural network**, however, is able to remember those characters because of its internal memory. It produces output, copies that output and loops it back into the network. The result of the process being a final output with a more accurate prediction.*

Let's now further use the following diagram to further understand how a recurrent neural network works.

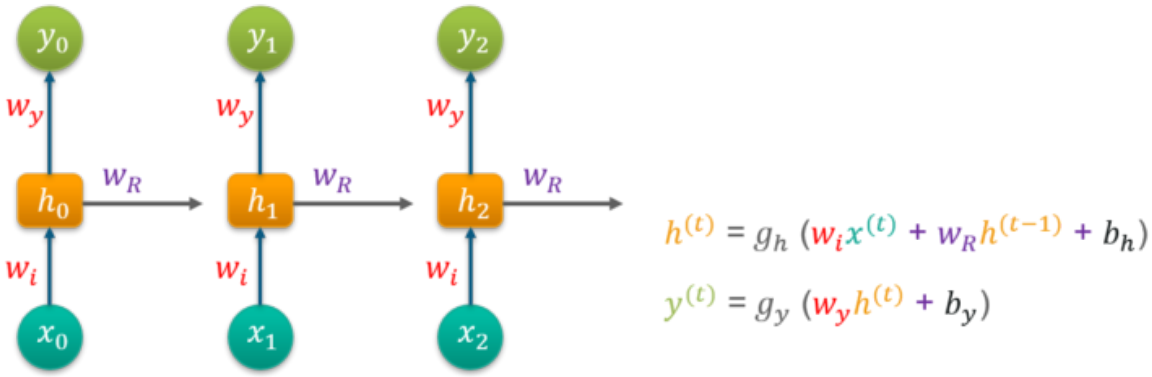


Image Source [Link]

We will consider **'w'** to be the weight matrix and **'b'** being the bias.

At time **t = 0**, the input is **'x0'** and the task is to figure out what is **'h0'**. Substituting **t = 0** in the equation and obtaining the function **h(t)** value. Next, the value of **'y0'** is found out using the previously calculated values when applied to the new formula.

This process is repeated through all of the timestamps in order to train a model. In addition, two algorithms, the ***back-propagation algorithm*** and the ***gradient descent algorithm*** are applied for every timestamp adjusting a recurrent neural network weights in order to reduce loss or the difference between actual and expected output, thereby optimising the neural network's performance.

What we've explained above is just a simple explanation of how neural networks work. A deeper understanding requires a mathematical run-through which is outside the scope of this article. You can read this article to get an in depth understanding [Link].

Benefits

The benefits of recurrent neural networks are as follows:

- The computation process takes into account historical information.
- The weights are shared across time.
- The model size may not necessarily increase with size of input.
- There is also a possibility of processing input of any length i.e. text data.

Limitations

Below are the drawbacks experienced when using recurrent neural networks:

- RNN experiences difficulty in memorising words from far away in the sequence and makes predictions based on only the most recent ones. This drawback is called the vanishing gradient problem. RNNs with the LSTM come in handy in such cases.
- RNN can sometimes assign high importance to the weights in the network without much reason. This problem is called exploding gradients. This problem is solved by truncating or squashing the gradients.
- RNNs cannot consider any future input for the current state.

Applications

- **Language modeling and text generation**
    - RNN's can predict the most suitable next character, next word, or next phrase in text.
- **Speech recognition**
    - RNN's can process audio recordings, parsed into acoustic signals then output syllables or phonetic elements matching each part of the recording.
- **Generating image description**
    - RNN's can take images, then identify important features in the image and generate brief texts that describe the image.
- **Time-series anomaly detection**
    - RNN's take sequential data series, such as the behavior of a user on a network over a full month and predicts which of the data in the series represents an anomaly compared to the normal flow of events.
- **Video tagging**
    - RNN's can take input, which is a series of video frames, and the model generates a textual description of what is happening in the video, frame by frame.

Types of Recurrent Neural Networks

Below are the different types of recurrent neural networks:

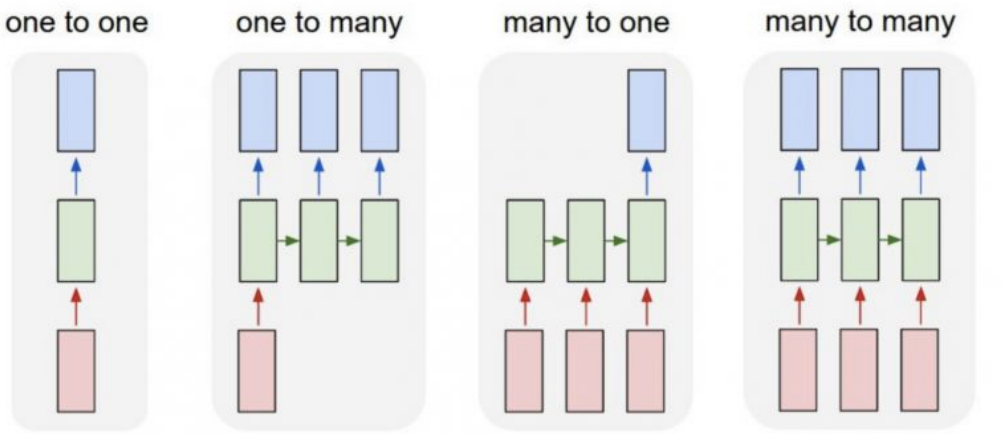
- **One-to-one RNN:** Also known as Vanilla Neural Network. This type of RNN is used for general machine learning problems, which has a single input and a single output.

- **One-to-many RNN:** This type of RNN has a single input and multiple outputs. An example of this is an image caption where the input is an image whereas the output is a sequence of words.

- **Many-to-one RNN:** This type of RNN takes a sequence of inputs (many inputs) and generates a single output. This structure is used when performing classification i.e. sentiment analysis where a sentence can be classified to sentence.

- **Many-to-many RNN:** This type of RNN takes a sequence of inputs (many inputs) and generates a sequence of outputs (many outputs). Machine translation is one of the examples.

The following diagram describes the above recurrent neural networks:



Source [Link]

RNN Architectures

- **Long Short-Term Memory (LSTM):** LSTM provides better performance by solving for the vanishing gradient problem, where the gradient gets smaller and smaller with each layer until it is too small to affect the deepest layers. They use a different function to compute the hidden state. In this way, they are capable of allowing the neural network to pick up pertinent information and save it, injecting it back into the model when necessary. Thus, LSTM remembers data from a long time ago.

- **Bidirectional RNNs:** These are two RNNs stacked on top of each other where the output computed is based on the hidden state of both RNNs. They assume that the correct output not only depends on the previous inputs in the time series but also on future inputs.

- **Deep RNNs:** Deep (Bidirectional) RNNs are similar to Bidirectional RNNs, only that we now have multiple layers per time step. This gives the neural network a higher learning capacity.

References

1. A Guide to RNN: Understanding Recurrent Neural Networks and LSTM. [Link]
2. Recurrent Neural Networks Cheatsheet. [Link]
3. Recurrent Neural Network [Link]
4. How Recurrent Neural Networks work [Link]
5. A Recurrent Neural Network Glossary: Uses, Types, and Basic Structure [Link]